IN THE SPECIFICATION

Please replace paragraph **0022** beginning on page 7 with the following marked up paragraph:

[0022] **Figure 2** illustrates a more detailed diagram of a processor, according to embodiments of the present invention. In particular, Figure 2 illustrates a more detailed diagram of one of processors 102/104 (hereinafter "processor 102"). As shown, memory interface unit 270 is coupled to cache buffers 256, register file 250 (that includes general purpose registers 252 and special purpose registers 254) and instruction buffer 202, such that memory interface unit 270 can retrieve macro instructions and associated operands and store such data into instruction buffer 202 and cache buffers 256, general purpose registers 252 and/or special purpose registers 254. Additionally, cache buffers 256 and register file 250 are coupled to decoder 204, functional units ~~212-218~~212, 214, 216, and 218 and retirement logic 228.

Please replace paragraph **0023** beginning on page 8 with the following marked up paragraph:

[0023] Decoder 204 is coupled to instruction buffer 202, such that decoder 204 retrieves the instructions from instruction buffer 202. Decoder 204 can receive these instructions and decode each of them to determine the given instruction and also to generate a number of instructions in an internal instruction set. For example, in one embodiment, the instructions received by decoder 204 are termed macro instructions, while the instructions that are generated by decoder 204 are termed micro instruction (or micro-operations). Decoder 204 is also coupled to instruction schedule 208, such that instruction scheduler 208 can receive these micro-operations for scheduled execution by functional units ~~212-218~~212, 214, 216, and 218.

Please replace paragraph **0024** beginning on page 8 with the following marked up paragraph:

[0024] Instruction scheduler 208 is coupled to dispatch logic 226, such that the instruction scheduler 208 transmits the instructions to be executed by functional units ~~212-218~~212, 214, 216, and 218. Dispatch logic 226 is coupled to functional units ~~212-218~~212, 214, 216, and 218 such that dispatch logic 226 transmits the instructions to functional units ~~212-218~~212, 214, 216, and 218 for execution. Functional units ~~212-218~~212, 214, 216, and 218 can be one of a number of different execution units, including, but not limited to, an integer arithmetic logic unit (ALU), a

floating-point unit, memory load/store unit, etc. Functional units ~~212-218~~212, 214, 216, and 218 are also coupled to retirement logic 228, such that functional units ~~212-218~~212, 214, 216, and 218 execute the instructions and transmit the results to retirement logic 228. Retirement logic 228 can transmit these results to memory that can be internal or external to processor 102, such as registers within register file 250 or cache buffers 256, or memory 132 (external to processor 102).

Please replace paragraph **0040** beginning on page 14 with the following marked up paragraph:

Accordingly, in an embodiment, translation unit 180 can translate the first binary into a different binary wherein at least one of the procedures are in-lined with the program code that called the procedure. For example, if the main procedure, "main()", included an invocation of a procedure "first_procedure(x,y)" wherein five lines of code are included within "first_procedure(x,y)", translation unit 180 can modify the binary such that the procedure call is removed and the five lines of code are included within "main()" directly. Accordingly, parameters x and y would not be placed on the stack. However, because this different binary is based on the second instruction set architecture as ~~relates~~related to dereferencing of the stack pointer subsequent to a return from a procedure call for a parameter within the procedure call, the program code will not include such de-referencing. As will be described in more detail below, because the binary is based on an instruction set architecture that ensures that the value "V" will not be accessed by location "L" in reference to the stack pointer subsequent to the completion of the procedure, hardware translation by decoder 204 can also be performed in conjunction with and/or exclusive of this software translation.

Please replace paragraph **0044** beginning on page 15 with the following marked up paragraph:

[0044] For example, upon receiving a floating-point multiply instruction, decoder 204 generates the micro-operations for the Intel® IA-64 instruction set architecture (instead of the micro-operations for the Intel® IA-32 instruction set architecture), thereby directing the associated floating-point unit (among functional units ~~212-218~~212, 214, 216, and 218) to modify the 80-bit operands to be 64-bit operands and to execute the floating-point multiply instruction as the associated instruction for the Intel® IA-64 instruction set architecture. Therefore, the precision of the floating-point operands will be reduced; however, the floating-point instructions

being based on the new instruction set architecture could increase performance in the execution of the application.

Please replace paragraph **0047** beginning on page 17 with the following marked up paragraph:

[0047] Therefore, in an embodiment, one of special purpose registers 254 can be employed as part of the hardware stack in addition to the stack within memory, such as memory 132, external to processor 102. Accordingly, this reduces the number of load and store operations by functional units ~~212-218~~212, 214, 216, and 218 association with the hardware stack. In particular, Figure 4 illustrates source code and the generated assembly code wherein a register is and is not employed as part of the hardware stack, according to embodiments of the present invention. As shown, Figure 4 includes source code 402, assembly code 404 and assembly code 406. Assembly code 404 includes portions of the assembly code instructions generated for source code 402 when a register within processor 102 is not employed as part of the hardware stack. Assembly code 406 includes portions of the assembly code instructions generated for source code 402 when a register within processor 102 is employed as part of the hardware stack.

Please replace paragraph **0060** beginning on page 21 with the following marked up paragraph:

[0060] Returning to flow diagram 300 of Figure 3, at process block 314, the instructions (that may have been modified as described above) are executed. In an embodiment, functional units ~~212-218~~212, 214, 216, and 218 execute the instructions. The software and hardware translations described herein are by way of example and not by way of limitation, as embodiments of the present invention can include other translations (both software and hardware) of a first binary based on a first instruction set architecture to a second binary based on a combination of the first instruction set architecture and a second instruction set architecture. Moreover, while a given translation has been described in reference to software or hardware, embodiments of the present invention are not so limited. For example, while a given translation has been described in relationship to a software translation, in another embodiment, such a translation can be performed in hardware and/or a combination of the hardware and software.